



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO.  | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|--|-------------|----------------------|---------------------|------------------|
| 10/788,801   | 02/27/2004  | Scott Musson         | ORACL-01382US0      | 7864             |
| 80548  | 7590        | 09/04/2009           |                     |                  |
| FLIESLER MEYER LLP<br>650 CALIFORNIA STREET<br>14TH FLOOR<br>SAN FRANCISCO, CA 94108 |             |                      |                     |                  |
| EXAMINER   |             |                      |                     |                  |
| HASSAN, RASHEDUL   |             |                      |                     |                  |
| ART UNIT   |             | PAPER NUMBER         |                     |                  |
| 2179   |             |                      |                     |                  |
| NOTIFICATION DATE  |             | DELIVERY MODE        |                     |                  |
| 09/04/2009   |             | ELECTRONIC           |                     |                  |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

OFFICEACTIONS@FDML.COM

### Office Action Summary

**Application No.**

10/788,801

**Applicant(s)**

MUSSON ET AL.

**Examiner**

RASHEDUL HASSAN

**Art Unit**

2179

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 05 August 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1,4-9,11-15,30,33-38,40-44 and 50-52 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,4-9,11-15,30,33-38,40-44 and 50-52 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date 7/7/2009.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

## **DETAILED ACTION**

### ***Continued Examination Under 37 CFR 1.114***

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 08/05/2009 has been entered.

### ***Claim Objections***

Claims 13 and 42 are objected to because of the following informalities: Claims 13 and 42 recite the limitation "**an** hypertext transfer protocol" (emphasis added), wherein "an" should be replaced by "a". Appropriate correction is required.

### ***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

**Claims 6, 13, 35 and 42 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.**

Claims 6 and 13 are recited to be dependent from cancelled claim 3, and claims

35 and 42 are recited to be dependent from cancelled claim 32. The examiner assumes for the sake of prosecution that the applicant intends to make claims 6 and 13 to be dependent from claim 1 and claims 35 and 42 to be dependent from claim 30.

Appropriate correction is required.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

**Claims 1, 4-9, 11-15, 30, 33-38, 40-44, and 50-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anuff et al. (US 6,327,628 B1) hereinafter Anuff, in view of Hough et al. (US 2002/0118226 A1) hereinafter Hough.**

For claims 1 and 30, Anuff teaches a computer implemented **method for supporting a portal application, comprising:**

**accepting a request, at a container on one or more web servers, from a user that interacts with a graphical user interface (GUI) of a web application at a client side** (e.g., accepting a request from a browser in a client device 10 sent to a portal web site, i.e., to a container, on a server device 12. See Fig. 1, c3:1-24, and also c4:15-45);

**mapping the request to a control tree factory** (e.g., to the process management services that are provided by a web server and suitable class libraries. See c4:15-45) **to generate a control tree** (e.g., the “control tree” is interpreted to mean relevant instantiated class objects implementing the requested GUI together with their interrelationships with each other as illustrated in Fig. 4 in Anuff), **where the control tree factory is independent of the container and is accessible from other containers** (e.g., the process management services that are provided by a web server and suitable libraries are independent of any particular web site and are accessible from other web sites too), **wherein at least one of the other containers is associated with at least one of a different protocol and a different application framework from the container** (e.g., Anuff mentions that both Java Server Pages (JSP) web sites or Active Server Pages (ASP) web sites can use the same Java libraries and services);

***generating the control tree in the container by the control tree factory based on the request*** (e.g., the “*control tree*” is interpreted to mean relevant instantiated class objects implementing the requested GUI together with their interrelationships with each other as illustrated in Fig. 4 in Anuff. Thus clearly these class objects are generated within the web site by the process management services mentioned above), ***wherein the control tree is a logical representation of the graphical user interface (GUI), wherein the control tree includes a set of controls that are related hierarchically to each other, and wherein each control of the set of controls represents one or more corresponding graphical and functional elements in the GUI of the web application;*** (The claim defines a “*control tree*” as “*a logical representation of the graphical user interface (GUI)*”. According to the instant disclosure, “*controls*” represent “corresponding graphical and functional elements in web applications ... In one embodiment, a control can be implemented as one or more classes in an object oriented programming paradigm”. Emphasis added, see [0028]. Therefore, “*a control*” is a “class” (in object oriented programming paradigm, hereinafter referred to as OOP) which is a logical representation of a corresponding graphical and functional element in a web application. In other words, the “*control tree*” is a collection of classes associated with a GUI since these classes can be seen as a logical representation of the GUI. Anuff teaches, with regard to Fig. 4, that back end controls/objects are related hierarchically to one another, e.g., A owns B and A is a subclass of B. Thus Anuff teaches generating a control tree wherein the control tree includes a set of controls which are related hierarchically to one another);

***advancing the control tree through at least one lifecycle stage in a sequence of one or more lifecycles*** (For a control, the lifecycle is defined in the instant disclosure, by a set of methods representing stages in the lifecycle. Life cycle stages are illustrated in Table 3 and appear to be nothing more than various stages of an object, instantiated from a class in the context of OOP, during runtime. Therefore, Anuff's controls for generating a portal GUI, implemented using objects in OOP, inherently advances the objects implementing the GUI through at least one lifecycle stage, e.g., at least the "Init" stage that allows a control to perform initialization based on interaction with each other in order to produce the response, i.e., the GUI, based on the request),

***aggregating the output of each control of the set of controls in the control tree to produce a response based on the request*** (e.g., Anuff implicitly teaches that the results of the processing output of each control is aggregated to produce the GUI of the portal application as illustrated in Fig. 2);

***providing the response to the container that contains the control tree*** (e.g., it is again implicit in the reference that the aggregated result of the underlying objects are delivered to the web site containing the objects); ***and***

***providing the response to the GUI of the web application at the client side*** (e.g., see the GUI of Fig. 2 provided to the client application).

Anuff however does not explicitly teach ***wherein at least one control in the control tree operates to interact with another control in the control tree through an event notification mechanism***. He explicitly teaches that an object model comprises a collection of objects that work together in documented relationships. See

Fig. 4. But, he does not explicitly teach that controls communicate through an event notification mechanism. However, in object oriented programming, communication/co-operation between objects using events was well known in the art at the time of the invention. Therefore, if not already implicitly taught by Anuff, it would have been obvious to a person of ordinary skill in the art to modify his invention so that controls can raise events and respond to events. The motivation for such modification would have been necessitated by the very nature of the GUI (portal) which is an interactive application and it is well known to a person of ordinary skill in the art that such applications are well suited for an event-driven implementation. For instance, Hough teaches a portal application wherein a portlet can raise a flag event to communicate with other portlets (see "Select and Flag User Interface Mechanism" section in page 5). the Examiner notes that the modules (i.e., portlets) in Anuff can be designed or implemented to perform any type of functionalities commonly known to a person of ordinary skill in the art at the time of the invention was made. In the same field of invention, Hough teaches a digital dashboard as a framework to build and deploy personalized portals that aggregate personal, team, corporate, and external information and services with single-click access to business intelligence and knowledge management functionality (See [0057]). Therefore, it would have been obvious to a person of ordinary skill in the art having the teaching of both Anuff and Hough before him/her at the time of the invention to implement a portal application including a plurality of portlets wherein each of the plurality of portlets is capable of communicating with another portlet of the plurality of portlets using event notification mechanism, since such a combination is not the result of novelty but of ordinary skill and common sense. Additionally, as Hough mentions, the



motivation for implementing communications within the plurality of portlets according to his invention would have been desirable to provide a system in which a user can easily join and filter information, to influence the relationship of information in different portlets of a portal application in a computer system (see Hough, [0014]).

For claims 4 and 33, Anuff further teaches ***wherein the step of generating a control tree comprises: creating a metadata representation of a control tree; and generating a class to construct the control tree based on the metadata representation.*** (See c6:34-46.)

For claims 5 and 34, Anuff further anticipates ***wherein the request is a hypertext transfer protocol request (HTTP); (See c6:57-58) and the request originates from a web browser.*** (See 16 in Fig. 1.)

For claims 6 and 35, Anuff further teaches ***providing the response to a web browser.*** (See Fig. 1, Fig. 2, c13:53-55)

For claims 7 and 36, Anuff further teaches ***wherein the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component.*** (Regarding an "interchangeable lifecycle component" the disclosure

mentions, in regard to Fig. 8, "The control container can use an interchangeable lifecycle driver 804 to drive the control tree through a sequence of states so that the request can be processed. As with the interchangeable persistence driver, an interface is provided to isolate lifecycle driver implementation details from the control container. This allows for different lifecycle implementations to be interchanged as needed". As for what constitutes the "interchangeable lifecycle driver/component", a reasonable interpretation would be, in absence of any explicit definition of the term in the disclosure and without importing limitations from the disclosure into the claim, to be objects/processes arbitrarily combined or divided into separate software, firmware or hardware components responsible to instantiate and carry out the run-time processing of the relevant classes/objects implementing the requested GUI which is inherent in Anuff.)

For claims 8 and 37, Anuff further teaches ***wherein each one of the set of controls can have an interchangeable persistence mechanism.*** (Anuff teaches object persistence using suitable database interface. See c4:16:32 and c5:45-48.)

For claims 9 and 38, Anuff further teaches ***wherein each one of the set of controls can render itself according to a theme.*** (See c8: 22-49.)

For claims 11 and 40, Anuff further teaches ***wherein one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls.*** (Since in OOP, objects can be instantiated in parallel and individually carry on their run-time processing in parallel with another object. Anuff also teaches multithreaded module preparation, c14:31-41.)

For claims 12 and 41, Anuff further teaches ***wherein a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.*** (Implicitly taught since objects apparently follow these stages in OOP which is well known to a person of ordinary skill in the art.)

For claims 13 and 42, Anuff further teaches ***wherein the response is an hypertext transfer protocol (HTTP) response.*** (See c6:61-65.)

For claims 15 and 44, Anuff further teaches ***wherein each one of the set of controls can be one of: Book, Page*** (see c4:65), ***Window, Menu, Layout*** (see c4:66), ***Portlet*** (modules, c4:65), ***Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.***

For claim 14 and 43, Anuff does not explicitly teach that controls can raise events and respond to events. However, this limitation has been already addressed in the rejections of claims 1 and 30 with respect to the event notification mechanism. Therefore, this claim is also obvious over Anuff in view of Hough based on the same rationale as already discussed in the rejection of claims 1 and 30 hereinabove.

For claim 50, Anuff further teaches that ***the one or more lifecycles of the control tree is provided and managed by the container and can be modified by the container*** (since, the controls in Anuff are provided and managed by the web site to which the controls belong).

For claim 51, Anuff further teaches ***wherein: each container associates a context object with the control tree factory, wherein each context object provide access to the protocol and application framework that is associated with the container*** (see "PortalPageContext" 34 in Fig. 4).

For claim 52, the combination further teaches ***the control tree factory uses one or more meta data to construct statically created controls at initialization of the control tree, wherein dynamically created controls are created in the control tree in reaction to state, context, and events during a control tree lifecycle*** (e.g., see

Anuff c4:33-35, wherein a default web site is provided containing statically created controls. It is well understood by one of ordinary skill in the art that the portal web site then can dynamically create other controls based on user interaction or generated events during the execution of the portal application).

### ***Response to Arguments***

Applicants' arguments filed 08/05/2009 have been fully considered but they are not persuasive.

Applicants have argued that Anuff and other cited prior art do not teach or make obvious that the control tree factory that generates the control tree is independent of the container and is accessible from multiple containers that are associated with different protocols and application frameworks. See pages 8-9 in the Remarks. The Examiner disagrees. Examiner interprets the "control tree factory" to the process management services that are provided by a web server and suitable class libraries as discussed in Anuff. See c4:15-45. A "container" has been interpreted as a "portal web site" as discussed in Anuff. Therefore, since the process management services that are provided by a web server and suitable libraries, are independent of any particular web site and are accessible from other web sites possibly associated with at least one of a different protocol and a different application framework (e.g., JSP or ASP), it follows that the limitations are taught by the Anuff reference.

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to RASHEDUL HASSAN whose telephone number is (571)272-9481. The examiner can normally be reached on M-F 7:30AM - 4PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Rashedul Hassan/  
Examiner, Art Unit 2179

/Weilun Lo/  
Supervisory Patent Examiner, Art Unit 2179